

RESEARCH

Open Access

# Collaborative emitter tracking using Rao-Blackwellized random exchange diffusion particle filtering

Marcelo G S Bruno<sup>1\*†</sup> and Stiven S Dias<sup>1,2†</sup>

## Abstract

We introduce in this paper the fully distributed, random exchange diffusion particle filter (ReDif-PF) to track a moving emitter using multiple received signal strength (RSS) sensors. We consider scenarios with both known and unknown sensor model parameters. In the unknown parameter case, a Rao-Blackwellized (RB) version of the random exchange diffusion particle filter, referred to as the RB ReDif-PF, is introduced. In a simulated scenario with a partially connected network, the proposed ReDif-PF outperformed a PF tracker that assimilates local neighboring measurements only and also outperformed a linearized random exchange distributed extended Kalman filter (ReDif-EKF). Furthermore, the novel ReDif-PF matched the tracking error performance of alternative suboptimal distributed PFs based respectively on iterative Markov chain move steps and selective average gossiping with an inter-node communication cost that is roughly two orders of magnitude lower than the corresponding cost for the Markov chain and selective gossip filters. Compared to a broadcast-based filter which exactly mimics the optimal centralized tracker or its equivalent (exact) consensus-based implementations, ReDif-PF showed a degradation in steady-state error performance. However, compared to the optimal consensus-based trackers, ReDif-PF is better suited for real-time applications since it does not require iterative inter-node communication between measurement arrivals.

**Keywords:** Distributed particle filters; RSS emitter tracking; Diffusion; Wireless sensor networks

## 1 Introduction

In several engineering applications, e.g., target tracking or fault detection, multiple agents [1] that are physically dispersed over remote nodes on a network cooperate to execute a global task, e.g., estimating a hidden signal or parameter, without relying on a global data fusion center. Each network node is normally equipped with one or more sensors that generate local measurements and can process those measurements independently of the rest of the network. At the same time, however, the network nodes are also able to communicate with each other in order to build in a collaborative fashion a joint estimate of the hidden signals or parameters of interest that depends both on local and remote measurements. Ideally, that joint estimate should be equal to or, at least

approximate the optimal global estimate that would be generated by a centralized processor with access to all network measurements.

Most of the previous literature in distributed signal processing on networks is based on linear estimation methods. Specifically, distributed versions of the Kalman filter were proposed e.g., in [2-4] to track unknown state vectors in linear, Gaussian state-space models. In situations, however, where the state dynamic model or the sensor observation models are nonlinear, the posterior distribution of the states conditioned on the network measurements becomes non-Gaussian (even with Gaussian sensor noise) and, therefore, the linear minimum mean square error (LMMSE) estimate of the states provided, e.g., by an extended Kalman filter (EKF) may differ from the true minimum mean square error (MMSE) estimate given by the expected value of the state vector conditioned on the measurements. In this paper in particular, we focus specifically on an application where multiple passive received-signal-strength (RSS) sensors jointly track

\*Correspondence: bruno@ita.br

†Equal contributors

<sup>1</sup>Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes 50, São José dos Campos, São Paulo 12228-900, Brazil

Full list of author information is available at the end of the article

a moving emitter assuming, at each network node, nonlinear observation models with possibly unknown static parameters.

### 1.1 Distributed particle filtering

In nonlinear scenarios, an alternative to approximate the true MMSE estimate is to use a sequential Monte Carlo method like particle filters [5,6]. Several distributed particle filters have been proposed recently, see a comprehensive review in [7], to handle nonlinear distributed estimation tasks. An important constraint in the design of a distributed estimation algorithm is, however, that most networks of practical interest are only partially connected, i.e., each node can only directly access neighboring nodes in its immediate vicinity according to the network topology. In particular, assuming conditional independence of the different sensor measurements given the state vector, a distributed particle filter (PF) normally requires the computation of a product of likelihood functions that depend on local data only [8]. To compute that product over the network in a fully distributed fashion and with local neighborhood inter-node communication only, previous references suggest using iterative average consensus [8], iterative Markov chain Monte Carlo move steps [9], or selective gossip algorithms [10]. Alternatively, we proposed in [11] to compute the likelihood product exactly in a finite number of iterations using either iterative minimum consensus [12] or flooding techniques [13]. However, both consensus or flooding-based solutions are very costly in terms of bandwidth requirements as they require multiple iterative inter-node communication between two consecutive sensor measurements. Previous works, e.g. [8,14,15], propose approximations aimed at reducing the communication cost, but, in all aforementioned schemes, processing and sensing at different time scales are still required.

### 1.2 Diffusion particle filtering

An alternative to circumvent the high communication cost of consensus algorithms is to use diffusion algorithms [16] which, contrary to the former, do not require multiple iterative inter-node communication between consecutive measurements. Diffusion algorithms are, however, sub-optimal in the sense that they do not simulate at each time step the behavior of the optimal global estimator, but rather, at best, approximate the optimal global solution asymptotically over time.

In the distributed linear estimation literature, most diffusion schemes are based on convex combinations of Kalman filters, see e.g., [3]. Kar et al. proposed in [2] a different approach based on random information dissemination. In a previous conference paper [17], we introduced the random exchange diffusion particle filter (ReDif-PF),

which generalizes and extends the methodology in [2] to a PF framework by basically using random information dissemination to build at each network node different Monte Carlo representations of the posterior distribution of the states conditioned on random sets of measurements coming from the entire network. Reference [17] assumed, however, that the parameters of the sensor observation model were perfectly known. In this paper, we extend the algorithm to a scenario with unknown parameters and derive in detail a Rao-Blackwellized [18] version of the ReDif-PF. In the specific application under consideration in the paper, the unknown parameters are the sensor variances, but most of the methodology in the derivation of the RB ReDif-PF is general and could be easily adapted to other signal models and applications provided that, in a fully Bayesian framework, the dynamic posterior probability distribution of the unknown parameters conditioned on the observations and on the simulated particles is a *conjugate prior* [19] for the likelihood function of the measurements.

An abbreviated description of the RB ReDif-PF may be found also in the short paper [20]. This paper consolidates and extends both [17] and [20] including detailed derivations and additional simulation results and comparisons. We also detail approximate versions of the RB ReDif-PF where we use Gaussian mixture models (GMM) [21] and moment-matching techniques inspired by [22] to reduce communication requirements.

### 1.3 Paper outline

The paper is divided into six sections and three appendices. Section 1 is the introduction. Section 2 describes the state and sensor models. Section 3 describes the centralized PF and also briefly reviews the equivalent broadcast, consensus, and flooding implementations introduced in [11]. Section 4 derives the ReDif-PF algorithm considering alternate scenarios with both known and unknown parameters. In the unknown parameter case, we derive in detail the Rao-Blackwellized version of the ReDif-PF and introduce approximate versions thereof that enable significant reductions in communication cost. The performance of the proposed algorithms is evaluated with simulated data in a realistic scenario with 25 sensors in Section 5. We compare the ReDif-PF algorithm in the unknown parameter scenario to the optimal centralized PF and its equivalent consensus implementations. In the known parameter case, we also compare the proposed ReDif-PF tracker to the Markov chain Monte Carlo distributed particle filter (MCDPF) in [9], to a linearized random exchange distributed EKF, which is a variation of the algorithm proposed in [2], and to a distributed bootstrap particle filter based on selective gossip as proposed in [23]. Finally, we present our conclusions in Section 6.

Appendices 1 and 2 show the proof of some key results in the paper, and Appendix 3 describes the ReDif-EKF algorithm used for comparison purposes in Section 5.

## 2 Problem setup

For simplicity of notation, we use lowercase letters in this paper to denote both random variables/vectors and real-valued samples of random variables/vectors with the proper interpretation implicit in context.

Without loss of generality, we assume that the emitter trajectory is described by the white noise acceleration model [24]

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{u}_n \quad (1)$$

where  $\mathbf{x}_n \triangleq [x_n \dot{x}_n y_n \dot{y}_n]^T$  is the hidden state vector at time step  $n$  consisting of the positions and velocities of the target's centroid respectively in dimensions  $x$  and  $y$ ;  $\mathbf{F}$  is the state transition matrix; and  $\{\mathbf{u}_n\}$  is a sequence of independent, identically distributed (i.i.d.) zero-mean Gaussian vectors with covariance matrix  $\mathbf{Q}$ . Matrices  $\mathbf{F}$  and  $\mathbf{Q}$ , parameterized by the sampling period  $T$  and the acceleration noise  $\sigma_{\text{accel}}^2$ , are detailed in [11,24].

### 2.1 Observation model

Let  $\mathcal{N}(m, \sigma^2)$  denote the Gaussian probability distribution with mean  $m$  and variance  $\sigma^2$  and denote by  $\mathcal{IG}(a, b)$  the inverse-gamma probability distribution with parameters  $a$  and  $b$ . The measurements  $z_{r,0:n} = \{z_{r,0}, \dots, z_{r,n}\}$  in decibels relative to one milliwatt (dBm) at the  $r$ th node of a network of  $R$  RSS sensors are modeled as

$$z_{r,n} = g_r(\mathbf{x}_n) + \sqrt{\sigma_r^2} v_{r,n}, \quad (2)$$

where  $v_{r,n} \sim \mathcal{N}(0, 1)$ ,  $\sigma_r^2 \sim \mathcal{IG}(\alpha, \beta)$ ,  $\forall r \in \mathcal{R} \triangleq \{1, \dots, R\}$ , and  $\{\mathbf{x}_0, \{\mathbf{u}_n\}, \{v_{r,n}\}, \{\sigma_r^2\}\}$  are mutually independent for all  $n \geq 0$  and for all  $r \in \mathcal{R}$ . The nonlinear function  $g_r(\cdot)$  in (2) is in turn given by [25]

$$g_r(\mathbf{x}) = P_0 - 10 \zeta_r \log \left( \frac{\|\mathbf{H}\mathbf{x} - \mathbf{x}_r\|}{d_0} \right), \quad (3)$$

where  $\mathbf{x}_r$  represents the  $r$ th sensor position,  $\|\cdot\|$  is the Euclidean norm,  $(P_0, d_0, \zeta_r)$  are known model parameters (see [25] for details), and  $\mathbf{H}$  is a  $2 \times 4$  projection matrix such that  $H(1, 1) = H(2, 3) = 1$  and  $H(i, j) = 0$  otherwise. We also denote by  $\mathcal{N}_r$  the set of nodes in the neighborhood of node  $r$ . The real-valued constants  $\{\alpha, \beta\}$  are the model's hyper-parameters.

Note that in (2), we take a fully Bayesian approach and model the unknown sensor noise variances  $\{\sigma_r^2\}$ ,  $r \in \mathcal{R}$ , as random variables that are mutually independent for  $s \neq r$  and identically distributed *a priori* with an inverse-gamma distribution.

### 2.2 Problem statement and goals

Let  $z_{1:R,0:n}$  denote the set  $\{z_{r,t}\}$  for all network nodes  $r = 1, \dots, R$  and all time instants  $t = 0, \dots, n$ . Given  $z_{1:R,0:n}$ , we want to compute the MMSE estimate

$$\hat{\mathbf{x}}_{n|n} = E\{\mathbf{x}_n | z_{1:R,0:n}\} \quad (4)$$

at each instant  $n \geq 0$ , where  $E\{\mathbf{x}_n | z_{1:R,0:n}\}$  denotes the conditional expectation of  $\mathbf{x}_n$  given  $z_{1:R,0:n}$ .

In the sequel, we first describe in Section 3 a recursive, centralized PF algorithm that approximates the desired global MMSE in (4) at each time step  $n$  in a scenario with *unknown* sensor variance scales  $\{\sigma_r^2\}$ . Next, we review in Section 3.1 two fully distributed algorithms that operate on a partially connected network and allow *exact* in-network computation of the state estimate in (4) without a global data fusion center and with inter-node communication limited to a node's immediate neighborhood according to the network topology. The network connectivity is described by a graph  $\mathcal{G} = (\mathcal{R}, \mathcal{E})$  where  $\mathcal{R} = \{1, \dots, R\}$  is the set of nodes and the graph has an edge  $(u, v) \in \mathcal{E}$ ,  $(u, v) \in \mathcal{R} \times \mathcal{R}$  if and only if nodes  $u$  and  $v$  can communicate directly with each other. The particular network graph used in the simulation scenarios in this paper is described in detail in Section 5.

Finally, we introduce in Section 4 a novel diffusion-based algorithm, which is also fully distributed and relies on local inter-node communication only specified as before by the network graph  $\mathcal{G}$  but, rather than yielding an identical estimate (4) at each node, obtains at each node  $r$  a suboptimal estimate

$$\hat{\mathbf{x}}_{r,n|n} = E\{\mathbf{x}_n | \mathcal{Z}_{r,0:n}\}, \quad (5)$$

where  $\mathcal{Z}_{r,0:n}$  is a *random subset* of  $z_{1:R,0:n}$ , which is different at each node  $r$  and includes measurements coming from random locations in the entire network, as opposed to measurements coming only from node  $r$  and its neighborhood. Compared to the exact distributed implementations of the optimal global estimate in Section 3.1, the diffusion solution in Section 4, although suboptimal, is designed to have a much lower inter-node communication cost and, therefore, is better suited for real-time applications.

## 3 Centralized particle filter

In a centralized architecture, all nodes in the network transmit their local measurements to a data fusion center which then runs a particle filter that approximates the MMSE estimate of the unknown state vector at each time instant  $n$  as

$$E\{\mathbf{x}_n | z_{1:R,0:n}\} \approx \sum_{q=1}^Q w_n^{(q)} \mathbf{x}_n^{(q)}, \quad (6)$$

where  $\{\mathbf{x}_n^{(q)}\}$ ,  $q \in \mathcal{Q} \triangleq \{1, \dots, Q\}$ , with the corresponding importance weights  $\{w_n^{(q)}\}$  is a *properly weighted* Monte Carlo set [5,6] that represents the posterior probability density function (PDF)  $p(\mathbf{x}_n|z_{1:R,0:n})$  in the sense that the sum on the right-hand side of (6) converges, according to some statistical criterion, to the expectation on the left-hand side when  $Q \rightarrow \infty$ . The random samples  $\{\mathbf{x}_n^{(q)}\}$ , also called particles, are sequentially generated according to a proposal probability distribution specified by a so-called importance PDF  $\pi(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(q)}, z_{1:R,0:n})$ . If the blind importance function [5]

$$\pi(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(q)}, z_{1:R,0:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)})$$

is used, then it turns out that the proper importance weights must be updated according to the recursion [6]

$$w_n^{(q)} \propto w_{n-1}^{(q)} p(z_{1:R,n}|\mathbf{x}_{0:n}^{(q)}, z_{1:R,0:n-1}) \quad (7)$$

where  $\propto$  denotes ‘proportional to,’  $z_{1:R,n}$  is an alternative notation for the set  $\{z_{r,n}\}$ ,  $r \in \mathcal{R}$ , and the proportionality constant on the right-hand side of (7) is chosen such that  $\sum_{q=1}^Q w_n^{(q)} = 1$ . From the mutual independence assumptions in the model in Section 2, it follows that

$$p(z_{1:R,0:n}|\mathbf{x}_{0:n}, \sigma_{1:R}^2) = \prod_{r=1}^R p(z_{r,0:n}|\mathbf{x}_{0:n}, \sigma_r^2) \quad (8)$$

and

$$\begin{aligned} p(\sigma_{1:R}^2|\mathbf{x}_{0:n}) &= \prod_{r=1}^R p(\sigma_r^2|\mathbf{x}_{0:n}) \\ &= \prod_{r=1}^R p(\sigma_r^2). \end{aligned} \quad (9)$$

From (8) and (9), it can be shown then that (see the proof in [11])

$$p(z_{1:R,n}|\mathbf{x}_{0:n}^{(q)}, z_{1:R,0:n-1}) = \prod_{r=1}^R \underbrace{p(z_{r,n}|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1})}_{\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)})}. \quad (10)$$

Substituting now (10) into (7), the centralized weight update rule reduces to

$$w_n^{(q)} \propto w_{n-1}^{(q)} \prod_{r=1}^R \underbrace{p(z_{r,n}|\mathbf{x}_{0:n}^{(q)}, z_{r,0:n-1})}_{\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)})}. \quad (11)$$

### 3.1 Equivalent distributed implementation of the centralized particle filter

Note that each factor  $\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)})$  in the product on the right-hand side of (11) depends only on local observations. In a fully connected network, assuming that all nodes  $r \in \mathcal{R}$  start out at instant  $n-1$  with the same

particles  $\{\mathbf{x}_{n-1}^{(q)}\}$ , they can all synchronously draw [26] new particles  $\{\mathbf{x}_n^{(q)}\}$  according to  $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(q)})$ , locally compute their own local likelihood functions  $\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)})$ , and then broadcast them to the entire network until all nodes have all the remote likelihood functions and can compute the product on the right-hand side of (11). Synchronous multinomial resampling according to the global weights followed by regularization may follow (see [11]) to mitigate particle degeneracy and impoverishment [5,6]. The algorithm described in this paragraph is referred to as the decentralized particle filter (DcPF) in [11] and [27].

As mentioned, however, in Section 1, real-world networks are only partially connected and fully distributed computations of the product in (11) are needed. One possibility is to approximate the product using iterative average consensus [28] as proposed, e.g., in [8] and [29]. Alternatively, we introduced in [11] a fully distributed computation of the global weights in (11) using either iterative minimum consensus [12] or flooding [13]. Both algorithms assume only local communication between nodes in immediate neighborhoods and, to achieve an *exact* computation of the global weights, require only a finite number of iterative message exchanges between nodes in the time interval between two consecutive sensor measurements.

Let  $D$  denote the diameter of the network graph, i.e., the maximum number of hops between any two nodes and, as before, denote by  $R$  the number of nodes in the network. By running  $R \times D$  consecutive minimum consensus iterations [12] for each particle  $q$ , it is possible (see details in [11]) to build an identical ordered list of likelihood functions  $\{\lambda_{r,n}^{(q)}(\mathbf{x}_n^{(q)})\}$ ,  $r \in \mathcal{R}$ , at all nodes. Each node can then locally compute the product of the likelihoods as in (10) and obtain identical, optimal global importance weights  $\{w_n^{(q)}\}$ . We refer to that (communication-intensive) minimum-consensus-based distributed tracking algorithm as CbPFs.

A more efficient way, however, to compute the exact optimal global weights at each node is to flood [13] the local node likelihoods over the network. Flooding protocols allow one to (iteratively) broadcast values over a network relying on local neighborhood inter-node communication only. Given a partially connected sensor network, one can simultaneously flood the  $R$  distinct likelihoods over the network as follows. First of all, each node  $r$  maintains an ordered list of distinct likelihoods. A likelihood in turn is flagged to indicate that it has not been sent to node  $r$  neighbors yet. Initially, the node  $r$  stores its local flagged likelihood in its list. At a given iteration, node  $r$  sends its lowest flagged likelihood to all neighbors and then unflags it. Conversely, it receives remote likelihoods from nodes  $s \in \mathcal{N}_r$ . If a received remote likelihood

is not included in node  $r$ 's list yet, it is inserted with a flag in its list. This procedure is guaranteed to converge in a finite number of iterations as soon as each node has  $R$  distinct values in its ordered list of likelihoods. We refer to the flooding-based iterative tracker in this paper as the CbPFb algorithm.

Figure 1 illustrates how the proposed flooding protocol iteratively creates at each node  $r$  an ordered list comprising all likelihoods across the network in a toy example with three nodes where node 1 is connected to node 2, node 2 is connected to nodes 1 and 3, and node 3 is connected to node 2 only. A star symbol is employed to indicate which likelihoods are flagged in the ordered list maintained by each node  $r$  at a given iteration  $j$ .

Although optimal in the sense of reproducing the centralized solution, the minimum consensus and flooding algorithms in [11] are still communication-intensive due to the requirement of iterative inter-node communication between sensor measurement arrivals. In the next sections, we describe an alternative fully distributed diffusion-based solution that drops this requirement and is the main topic of this paper.

#### 4 Random exchange diffusion particle filter

In this section, we derive an alternative distributed PF based on random information dissemination that extends the methodology in [2] to a Monte Carlo framework. We also present a Rao-Blackwellized version of the proposed distributed PF in a scenario with unknown sensor parameters.

Let  $\mathcal{Z}_{s,0:n-1}$  denote the set of all network measurements assimilated by node  $s$  up to instant  $n-1$ . Next, let  $\{\mathbf{x}_{s,0:n-1}^{(q)}\}$  with associated weights  $\{w_{s,n-1}^{(q)}\}$ ,  $q \in \mathcal{Q}$ , be a properly weighted set that represents the posterior PDF

$p(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})$  at node  $s$ . Assume now that at instant  $n-1$ , node  $s$  sends its particles and weights to a neighboring node  $r$  that can assimilate at instant  $n$  the measurements  $\mathcal{Z}_{r,n} = \{z_{i,n}\}$ ,  $i \in \{r\} \cup \mathbf{N}_r$ . At instant  $n$ , the new particle set at node  $r$ ,  $\mathbf{x}_{r,0:n}^{(q)} = (\mathbf{x}_{s,0:n-1}^{(q)}, \mathbf{x}_{r,n}^{(q)})$  with updated weights  $w_{r,n}^{(q)}$  such that

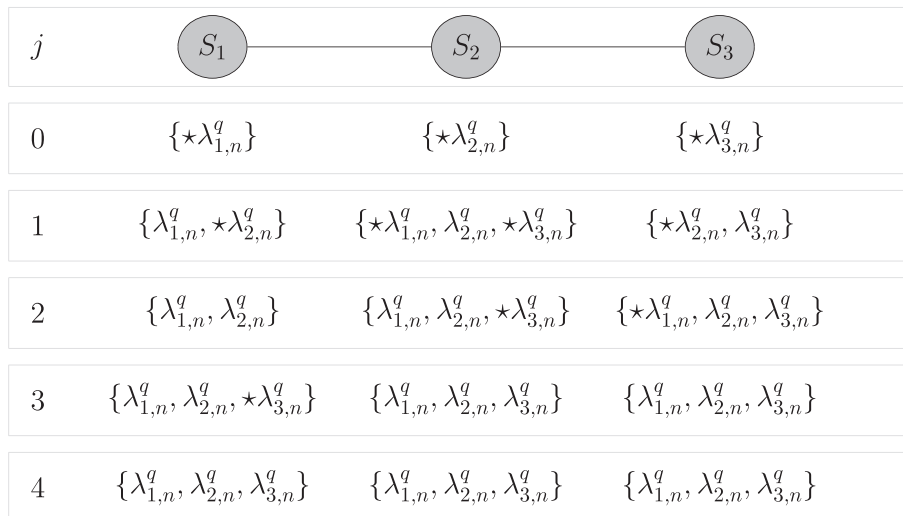
$$\mathbf{x}_{r,n}^{(q)} \sim p(\mathbf{x}_n|\mathbf{x}_{s,n-1}^{(q)}) \quad (12)$$

$$w_{r,n}^{(q)} \propto w_{s,n-1}^{(q)} p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1}) \quad (13)$$

is now a properly weighted set to represent the updated posterior  $p(\mathbf{x}_{0:n}|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1})$ , where  $\{\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}\}$  is redefined as  $\mathcal{Z}_{r,0:n}$ . Resampling from the particle weights followed by regularization may be added to combat particle degeneracy and restore particle diversity, i.e., for  $q \in \mathcal{Q}$  (see also [11]):

- Draw  $l^{(q)}$  from  $\{1, 2, \dots, Q\}$  with  $Pr(\{l^{(q)} = l\}) = w_{r,n}^{(l)}$ , where  $Pr(A)$  denotes the probability of an event  $A$ .
- Make  $\tilde{\mathbf{x}}_{r,n}^{(q)} = \mathbf{x}_{r,n}^{(l^{(q)})} + hD_n\mathbf{x}^*$ , where  $\mathbf{x}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ,  $D_n D_n^T$  is equal to the empirical covariance of the weighted particles  $\{\mathbf{x}_{r,n}^{(q)}, w_{r,n}^{(q)}\}$ , and  $h > 0$  is an empirically adjusted parameter.
- Reset the particle weights  $w_{r,n}^{(q)}$  to  $\frac{1}{Q}$  and make  $\mathbf{x}_{r,n}^{(q)} = \tilde{\mathbf{x}}_{r,n}^{(q)}$ .

**Random exchange protocol** In order to build, at each instant  $n$  and at each node  $r$ , different Monte Carlo representations of the posterior distribution conditioned on different sets of observations  $\mathcal{Z}_{r,0:n}$  coming from random



**Figure 1** Likelihood flooding protocol. Illustrative example in a linear network with three nodes.

locations in the entire network, it suffices to implement a protocol where each node  $r$ , starting from instant zero, exchanges its particles and weights with a randomly chosen neighboring node  $s$ , propagates the received particles using the blind importance function as in (12), and then updates their weights as in (13).

Figure 2 illustrates the evolution of the marginal posterior at each node - in a linear network containing three nodes running the random exchange protocol - over four time instants. Initially, each node  $r \in \{1, 2, 3\}$  has a posterior at instant zero conditioned on the measurements  $\mathcal{Z}_{r,0} = \{z_{i,0}\}$ ,  $i \in \{r\} \cup \mathbf{N}_r$ , in its vicinity only. At each time instant  $n \in \{1, 2, 3\}$ , network nodes perform the sequence of random exchanges as indicated in the rightmost column of Figure 2 and, then, update the received posterior by assimilating measurements in their respective neighborhoods.

Note that in the linear network topology shown in Figure 2, node 2 always performs two random exchanges at each time instant  $n$ . Generally speaking, however, at a given instant  $n$ , a node  $r$  exchanges its parameters at least one time with a randomly chosen neighbor  $s$  and, in the worst case, performs  $d(r)$  random exchanges between two measurement arrivals with nodes in its vicinity, where  $d(r)$  is the degree of node  $r$ , i.e., the number of neighbor nodes.

Unlike randomized gossip algorithms [30], this procedure diffuses information by randomly propagating posterior statistics across the network. More specifically, as the initial posterior statistics provided by a given node  $r_0$  at time 0 follows a path  $\mathcal{P} \triangleq \{r_0, r_1, \dots, r_n\}$  along the network, it assimilates the available measurements  $\mathcal{Z}_{r,n}$  in the neighborhood of each visited node  $r \in \mathcal{P}$ . Since, as illustrated in Figure 2, the initial posteriors at each node follow different paths, the posterior available

at node  $r_n$  at time  $n$  will be different from those in the remaining nodes. Thus, network nodes will provide different estimates conditioned on distinct sets of measurements.

#### 4.1 ReDif-PF with known sensor variances

If the parameters of the sensor observation model at each node  $r$  are deterministic and perfectly known, then

$$p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1}) = \prod_{i \in \{r\} \cup \mathbf{N}_r} p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}). \quad (14)$$

At instant  $n$ , then, upon receiving  $(w_{s,n-1}^{(q)}, \mathbf{x}_{s,n-1}^{(q)})$ ,  $q \in \mathcal{Q}$ , from node  $s$ , the particle filter at node  $r$  samples as before

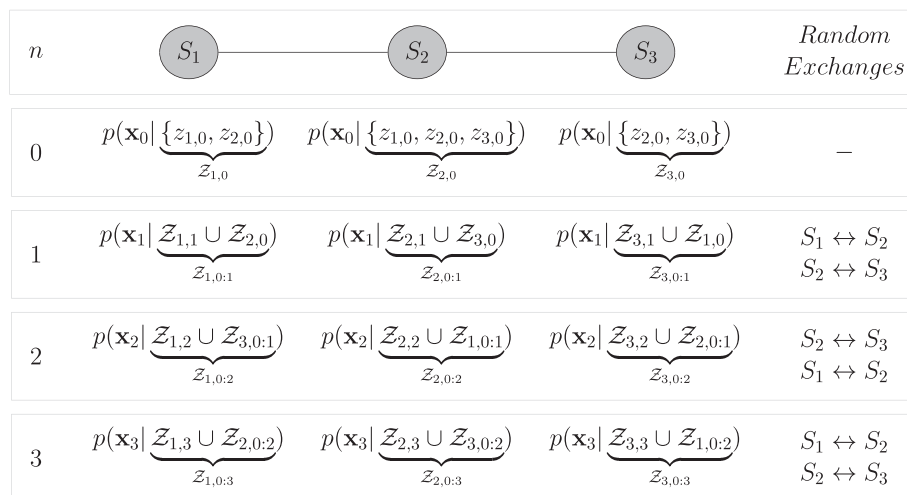
$$\mathbf{x}_{r,n}^{(q)} \sim p(\mathbf{x}_n | \mathbf{x}_{s,n-1}^{(q)}) \quad q = 1, \dots, Q$$

and updates its weights as

$$w_{r,n}^{(q)} \propto w_{s,n-1}^{(q)} \prod_{i \in \{r\} \cup \mathbf{N}_r} p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}) \quad q = 1, \dots, Q \quad (15)$$

where  $z_i | \mathbf{x}_{r,n}^{(q)} \sim \mathcal{N}(g_i(\mathbf{x}_{r,n}^{(q)}), \sigma_i^2)$ .

**Inter-node transmission requirements** From the previous discussion, it follows that in the scenario with known variances at each instant  $n$ , it suffices for each node  $s$  to transmit to the chosen neighbor  $r$  the set of particles  $\{\mathbf{x}_{s,n-1}^{(q)}\}$  ( $4Q$  real numbers for a four-dimensional state space) and the respective set of importance weights  $\{w_{s,n-1}^{(q)}\}$  ( $Q$  real numbers). In addition, node  $s$  also sends its scalar observation  $z_{s,n}$  and the known observation model parameters  $(\zeta_s, \mathbf{x}_s, \sigma_s^2)$  (see (3)) to all nodes  $i$  in the neighborhood of  $s$ .



**Figure 2 Random exchange protocol.** Evolution of the marginal posterior at each node in a linear network with three nodes.

#### 4.2 Rao-Blackwellized ReDif-PF with unknown sensor variances

Let  $IG(\sigma^2|\alpha, \beta)$  denote the PDF of a continuous random variable  $\sigma^2$  with an inverse-gamma distribution specified by the parameters  $\alpha$  and  $\beta$ , i.e. [19],

$$IG(\sigma^2|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \sigma^{-2(\alpha+1)} \exp\left(-\frac{\beta}{\sigma^2}\right) \quad \sigma^2 > 0 \quad (16)$$

and zero otherwise. In (16),  $\Gamma(\cdot)$  denotes the gamma function

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} \exp(-t) dt \quad \alpha > 0.$$

Similarly, let also  $N(\mathbf{x}|\mathbf{m}, \mathbf{\Sigma})$  denote the PDF of a Gaussian random vector taking values in  $\mathbb{R}^L$  and with mean  $\mathbf{m}$  and positive definite covariance matrix  $\mathbf{\Sigma}$ , i.e.,

$$N(\mathbf{x}|\mathbf{m}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{L}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{(\mathbf{x}-\mathbf{m})^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\mathbf{m})}{2}\right],$$

where  $|\mathbf{\Sigma}|$  denotes the determinant of the matrix  $\mathbf{\Sigma}$  and the superscript T denotes the transpose of a vector.

In the scenario with *unknown* sensor variances, it can be shown (see Appendix 2) that if at instant  $n-1$ ,

$$p(\sigma_{1:R}^2|\mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) = \prod_{i=1}^R IG(\sigma_i^2|\alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}), \quad (17)$$

then

$$p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1}) = \prod_i \underbrace{p(z_{i,n}|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1})}_{\tilde{\lambda}_{i,n}^{(q)}(\mathbf{x}_{r,n}^{(q)})}, \quad (18)$$

where  $i \in \{r\} \cup \mathbf{N}_r$ , and each factor  $\tilde{\lambda}_{i,n}^{(q)}(\mathbf{x}_{r,n}^{(q)})$  in the product on the right-hand side of (18) is computed by solving the integral

$$\begin{aligned} & \int_0^\infty p(z_{i,n}|\mathbf{x}_{r,n}^{(q)}, \sigma_i^2) p(\sigma_i^2|\mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) d\sigma_i^2 \\ &= \int_0^\infty N(z_{i,n}|g_i(\mathbf{x}_{r,n}^{(q)}), \sigma_i^2) IG(\sigma_i^2|\alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) d\sigma_i^2 \\ &\propto \frac{[\beta_{s,i,n-1}^{(q)}]^{\alpha_{s,i,n-1}}}{\Gamma(\alpha_{s,i,n-1})} \frac{\Gamma(\alpha_{r,i,n})}{[\beta_{r,i,n}^{(q)}]^{\alpha_{r,i,n}}}, \end{aligned} \quad (19)$$

where  $\Gamma(\cdot)$ , as before, denotes the gamma function

$$\alpha_{r,i,n} = \alpha_{s,i,n-1} + \frac{1}{2} \quad (20)$$

$$\beta_{r,i,n}^{(q)} = \beta_{s,i,n-1}^{(q)} + \frac{1}{2} [z_{i,n} - g_i(\mathbf{x}_{r,n}^{(q)})]^2, \quad (21)$$

with  $g_i(\cdot)$  calculated as in (3). Furthermore, at node  $r$  and instant  $n$ , the updated parameter posterior PDF

$$p(\sigma_{1:R}^2|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{r,0:n}) = \prod_{i=1}^R IG(\sigma_i^2|\alpha_{r,i,n}, \beta_{r,i,n}^{(q)}), \quad (22)$$

where  $\alpha_{r,i,n}$  and  $\beta_{r,i,n}^{(q)}$  are updated as in (20) and (21) if  $i \in \{r\} \cup \mathbf{N}_r$  or, otherwise, are kept equal respectively to  $\alpha_{s,i,n-1}$  and  $\beta_{s,i,n-1}^{(q)}$ . If regularization is used to combat particle degeneracy, the posterior parameters  $\{\beta_{s,i,n-1}^{(q)}\}$  must be also resampled according to new weights  $w_{r,n}^{(q)}$  updated as in (13) and a new set  $\{\beta_{r,i,n}^{(q)}\}$  must be recalculated for  $i \in \{r\} \cup \mathbf{N}_r$  using (21) with the resampled  $\{\beta_{s,i,n-1}^{(q)}\}$  and the new moved particles  $\{\mathbf{x}_{r,n}^{(q)}\}$ . We follow, however, a different suboptimal strategy described in Section 4.3, which also allows a significant reduction in inter-node communication cost.

**Inter-node transmission requirements** In the unknown variance scenario, based on the previous discussion, at each instant  $n$ , a node  $s$  has to transmit to its (randomly chosen) neighboring node  $r$  its particle set  $\{\mathbf{x}_{s,n-1}^{(q)}\}$  ( $4Q$  real numbers) plus the respective importance weights  $\{w_{s,n-1}^{(q)}\}$  ( $Q$  real numbers) and the set of hyper-parameters  $\{(\alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)})\}$ ,  $i \in \mathcal{R}$ ,  $q \in \mathcal{Q}$  (another  $R \times (Q+1)$  real numbers), which specify the posterior PDF  $p(\sigma_{1:R}^2|\mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$ . In addition, as before, node  $s$  also sends its scalar observation  $z_{s,n}$  and the observation model parameters  $(\zeta_s, \mathbf{x}_s)$  to all nodes  $i$  in the neighborhood of  $s$ .

#### 4.3 Approximate RB ReDif-PF

Although the exact ReDif-PF algorithms in Sections 4.1 and 4.2 converge asymptotically to the state estimate in (5) as the number of particles  $Q$  goes to infinity, their inter-node communication cost is still relatively high. To reduce the communication burden, we propose two *sub-optimal* approximations which are described in detail in the sequel.

**GMM approximation of the marginal posterior of the states** To circumvent the inconvenience of having to transmit, either in the known or unknown sensor param-

eter scenarios,  $Q$  particles and respective weights per node at each time step, we follow the lead in [21] and build a GMM representation of the marginal posterior  $p(\mathbf{x}_{n-1} | \mathcal{Z}_{s,0:n-1})$  of the form

$$p(\mathbf{x}_{n-1} | \mathcal{Z}_{s,0:n-1}) \approx \sum_{k \in \mathcal{K}} \eta_{s,n-1}^{(k)} N(\mathbf{x}_{n-1} | \boldsymbol{\mu}_{s,n-1}^{(k)}, \boldsymbol{\Sigma}_{s,n-1}^{(k)}) \quad (23)$$

where  $\mathcal{K} = \{1, \dots, K\}$  and the parameters  $\eta_{s,n-1}^{(k)}$ ,  $\boldsymbol{\mu}_{s,n-1}^{(k)}$ , and  $\boldsymbol{\Sigma}_{s,n-1}^{(k)}$  are obtained from the weighted particle set  $\{\mathbf{x}_{s,n-1}^{(q)}, w_{s,n-1}^{(q)}\}$ ,  $q \in \mathcal{Q}$ , at node  $s$  using the Expectation-Maximization (EM) [31] algorithm. Node  $s$  now transmits to node  $r$  only the parameters that specify the GMM model, i.e.,  $15K$  real numbers for a four-dimensional state vector, as opposed to  $5Q$  real numbers, where typically  $Q \gg K$  (in the simulations in Section 5 for example,  $K$  is either 1 or 2, whereas  $Q$  is 500). Node  $r$  then locally resamples  $Q$  new particles  $\mathbf{x}_{s,n-1}^{(q)}$  according to the received GMM PDF and resets its importance weights  $w_{s,n-1}^{(q)}$  to  $1/Q$ . Since resampling from the GMM approximation is used, we omit the regularization step mentioned in Section 4.

**Approximation of the posterior distribution of the sensor variances** In the particular situation where the sensor variances are unknown, in theory we should also locally resample the previous particle trajectories  $\mathbf{x}_{s,0:n-2}^{(q)}$  jointly with  $\mathbf{x}_{s,n-1}^{(q)}$  from some parametric approximation to  $p(\mathbf{x}_{0:n-1} | \mathcal{Z}_{s,0:n-1})$  and then recompute retroactively the posterior PDF's  $p(\sigma_i^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$ ,  $i = 1, \dots, R$  for the resampled particle paths. To eliminate that curse of dimensionality, it is desirable to introduce a parametric approximation to  $p(\sigma_i^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$  that eliminates the dependence of that function on the particle label  $q$  and the simulated sequence  $\mathbf{x}_{s,0:n-1}^{(q)}$ .

Specifically, we follow the lead in [11,22,32], and, for each  $i \in \mathcal{R}$ , approximate the marginal posteriors  $p(\sigma_i^2 | \mathbf{x}_{0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$  for all particle labels  $q$  and all possible sequences  $\mathbf{x}_{0:n-1}^{(q)}$  by a new inverse gamma PDF with parameters  $\tilde{\alpha}_{s,i,n-1}$  and  $\tilde{\beta}_{s,i,n-1}$ , independent of  $q$  and chosen such that the approximated PDF  $IG(\sigma_i^2 | \tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1})$  matches the first and second moments of

$$p(\sigma_i^2 | \mathcal{Z}_{s,0:n-1}) = \int [p(\sigma_i^2 | \mathbf{x}_{0:n-1}, \mathcal{Z}_{s,0:n-1}) \times p(\mathbf{x}_{0:n-1} | \mathcal{Z}_{s,0:n-1})] d\mathbf{x}_{0:n-1} \quad (24)$$

where the term on the left-hand side of (24) is the average (or *expected value*) of  $p(\sigma_i^2 | \mathbf{x}_{0:n-1}, \mathcal{Z}_{s,0:n-1})$  over all possible realizations of  $\mathbf{x}_{0:n-1}$  conditioned on the observations

$\mathcal{Z}_{s,0:n-1}$ . Assuming now that  $\{(w_{s,n-1}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)})\}$ ,  $q \in \mathcal{Q}$ , is a properly weight set available at node  $s$  at instant  $n-1$  to represent  $p(\mathbf{x}_{0:n-1} | \mathcal{Z}_{s,0:n-1})$ , we make the Monte Carlo approximation

$$p(\sigma_i^2 | \mathcal{Z}_{s,0:n-1}) \approx \sum_{q=1}^Q w_{s,n-1}^{(q)} p(\sigma_i^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}). \quad (25)$$

On the other hand, from the assumption that  $p(\sigma_{1:R}^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$  is a separable function factored as in (17), it follows that

$$p(\sigma_i^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) = IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)})$$

and, therefore,

$$p(\sigma_i^2 | \mathcal{Z}_{s,0:n-1}) \approx \sum_{q=1}^Q w_{s,n-1}^{(q)} IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}). \quad (26)$$

In the sequel, recall that if  $\sigma^2 \sim IG(\alpha, \beta)$ , then the respective mean and variance of  $\sigma^2$  are given by [19]

$$E\{\sigma^2\} = \frac{\beta}{\alpha - 1}, \quad \alpha > 1 \quad (27)$$

$$\text{Var}\{\sigma^2\} = \frac{\beta^2}{(\alpha - 1)^2 (\alpha - 2)}, \quad \alpha > 2. \quad (28)$$

Therefore, the parameters  $\tilde{\alpha}_{s,i,n-1}$  and  $\tilde{\beta}_{s,i,n-1}$  such that  $IG(\sigma_i^2 | \tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1})$  matches the mean and variance associated with the PDF on the right-hand side of (26) are found, following the procedure in [11,22,32] by making

$$\tilde{\alpha}_{s,i,n-1} = 2 + \widehat{E}_{s,n-1}^2[\sigma_i^2] / \widehat{\text{VAR}}_{s,n-1}[\sigma_i^2] \quad (29)$$

$$\tilde{\beta}_{s,i,n-1} = (\tilde{\alpha}_{s,i,n-1} - 1) \widehat{E}_{s,n-1}[\sigma_i^2], \quad (30)$$

where

$$\begin{aligned} \widehat{E}_{s,n-1}[\sigma_i^2] &= \frac{\sum_{q=1}^Q w_{s,n-1}^{(q)} \beta_{s,i,n-1}^{(q)}}{\alpha_{s,i,n-1} - 1} \\ \widehat{\text{VAR}}_{s,n-1}[\sigma_i^2] &= \frac{\sum_{q=1}^Q w_{s,n-1}^{(q)} (\beta_{s,i,n-1}^{(q)})^2}{(\alpha_{s,i,n-1} - 1)(\alpha_{s,i,n-1} - 2)} \\ &\quad - \widehat{E}_{s,n-1}^2[\sigma_i^2]. \end{aligned} \quad (31)$$



Replacing now  $p(\sigma_i^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$  in (19) with

$$\tilde{p}(\sigma_i^2 | \mathcal{Z}_{s,0:n-1}) = IG(\sigma_i^2 | \tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1})$$

for all  $q \in \{1, \dots, Q\}$  and *all possible sequences*  $\mathbf{x}_{s,0:n-1}^{(q)}$ , we get, at node  $r$  at instant  $n$ , new factors  $\tilde{\lambda}_{i,n}(\cdot)$  such that

$$\begin{aligned} \tilde{\lambda}_{i,n}(\mathbf{x}_{r,n}^{(q)}) &= \int_0^\infty p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) \tilde{p}(\sigma_i^2 | \mathcal{Z}_{s,0:n-1}) d\sigma_i^2 \\ &= \int_0^\infty N(z_{i,n} | g_i(\mathbf{x}_{r,n}^{(q)}, \sigma_i^2)) IG(\sigma_i^2 | \tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1}) d\sigma_i^2 \\ &\propto \frac{[\tilde{\beta}_{s,i,n-1}]^{\tilde{\alpha}_{s,i,n-1}}}{\Gamma(\tilde{\alpha}_{s,i,n-1})} \frac{\Gamma(\alpha_{r,i,n})}{[\beta_{r,i,n}^{(q)}]^{\alpha_{r,i,n}}}, \end{aligned} \quad (32)$$

where

$$\alpha_{r,i,n} = \tilde{\alpha}_{s,i,n-1} + \frac{1}{2} \quad (33)$$

$$\beta_{r,i,n}^{(q)} = \tilde{\beta}_{s,i,n-1} + \frac{1}{2} [z_{i,n} - g_i(\mathbf{x}_{r,n}^{(q)})]^2, \quad (34)$$

for all  $q \in \mathcal{Q}$  and all  $i \in \{r\} \cup \mathbf{N}_r$ . Otherwise, if  $i \notin \{r\} \cup \mathbf{N}_r$

$$\alpha_{r,i,n} = \tilde{\alpha}_{s,i,n-1} \quad (35)$$

$$\beta_{r,i,n}^{(q)} = \tilde{\beta}_{s,i,n-1}, \quad (36)$$

again for all  $q \in \mathcal{Q}$ . The modified importance weight update rule at node  $r$  at instant  $n$  now becomes

$$w_{r,n}^{(q)} \propto w_{s,n-1}^{(q)} \prod_{i \in \{r\} \cup \mathbf{N}_r} \tilde{\lambda}_{i,n}(\mathbf{x}_{r,n}^{(q)}) \quad q \in \{1, \dots, Q\}. \quad (37)$$

**Inter-node communication cost** By combining the GMM approximation and the moment-matching approximation described before, node  $s$  now transmits to its (randomly chosen) neighbor  $r$  only the GMM model parameters (15K real numbers as previously explained) plus  $2R$  hyper-parameters  $(\tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1})$ ,  $i \in \mathcal{R}$ , as opposed to  $R \times (Q+1)$  hyper-parameters as before in the exact RB ReDif-PF algorithm.

**Summary of the approximate RB-ReDif-PF** Algorithm 1 summarizes the approximate RB ReDif-PF tracker at node  $r$  at instant  $n$ . In Algorithm 1, the symbol  $\Theta_{r,n}$  denotes the set  $\left\{ \left( \eta_{r,n}^{(k)}, \boldsymbol{\mu}_{r,n}^{(k)}, \boldsymbol{\Sigma}_{r,n}^{(k)} \right), \left( \tilde{\alpha}_{r,i,n} \tilde{\beta}_{r,i,n} \right) \right\}$  for  $i \in \mathcal{R}$  and  $k \in \mathcal{K}$ .

**Algorithm 1** Approximate Rao-Blackwellized random exchange diffusion particle filter

- 
- 1: **procedure** REDIF-PF( $z_{r,n}, \Theta_{s,n-1}$ )
  - 2:   Send  $z_{r,n}$  to neighbors  $i \in \mathbf{N}_r$
  - 3:   Block until receive all  $z_{i,n}$  from nodes  $i \in \mathbf{N}_r$
  - 4:   Extract  $\eta_{s,n-1}^{(k)}, \boldsymbol{\mu}_{s,n-1}^{(k)}, \boldsymbol{\Sigma}_{s,n-1}^{(k)}$  from  $\Theta_{s,n-1}$ ,  $k \in \mathcal{K}$  and resample  $\mathbf{x}_{s,n-1}^{(q)}$ , for all  $q \in \mathcal{Q}$  from the GMM approximation defined by those parameters.
  - 5:   Extract  $(\tilde{\alpha}_{s,i,n-1}, \tilde{\beta}_{s,i,n-1})$ ,  $i \in \mathcal{R}$ , from  $\Theta_{s,n-1}$
  - 6:   **for**  $q \leftarrow 1$  **to**  $Q$  **do**
  - 7:     Sample  $\tilde{\mathbf{x}}_{r,n}^{(q)} \sim p(\mathbf{x}_n | \mathbf{x}_{s,n-1}^{(q)})$
  - 8:     Compute  $\alpha_{r,i,n}$  and  $\beta_{r,i,n}^{(q)}$ , for all  $i \in \mathcal{R}$  using (33) to (36).
  - 9:     Compute  $\tilde{w}_{r,n}^{(q)} \propto \prod_{i \in \{r\} \cup \mathbf{N}_r} \tilde{\lambda}_{i,n}(\tilde{\mathbf{x}}_{r,n}^{(q)})$  where  $\tilde{\lambda}_{i,n}(\tilde{\mathbf{x}}_{r,n}^{(q)})$  is given by (32).
  - 10:   **end for**
  - 11:   Normalize  $\tilde{w}_{r,n}^{(q)}$  such that  $\sum_q \tilde{w}_{r,n}^{(q)} = 1$ .
  - 12:   Make  $\hat{\mathbf{x}}_{r,n|n} = \sum_{q=1}^Q \tilde{w}_{r,n}^{(q)} \tilde{\mathbf{x}}_{r,n}^{(q)}$
  - 13:   Compute  $\tilde{\alpha}_{r,i,n}$  and  $\tilde{\beta}_{r,i,n}$  from the weighted particle set  $\{(\tilde{w}_{r,n}^{(q)}, \tilde{\mathbf{x}}_{r,n}^{(q)})\}$  using (29) and (30) for all  $i \in \mathcal{R}$ .
  - 14:   Compute the parameters  $\eta_{r,n}^{(k)}, \boldsymbol{\mu}_{r,n}^{(k)}$ , and  $\boldsymbol{\Sigma}_{r,n}^{(k)}$  of the GMM approximation to  $p(\mathbf{x}_n | \mathcal{Z}_{r,0:n})$  using the EM algorithm.
  - 15:   Exchange  $\Theta_{r,n}$  with  $\Theta_{s,n}$  from a randomly chosen neighbor  $s$  using the random exchange protocol, see [17].
  - 16:   **return**  $\hat{\mathbf{x}}_{r,n|n}, \Theta_{s,n}$ .
  - 17: **end procedure**
- 

#### 4.4 Differences between ReDif-PF and the Markov chain distributed particle filter

An alternative and different approach to distributed particle filtering is the MCDPF algorithm introduced in [9]. MCDPF, like other previous work in the distributed PF literature, assumes conditional independence of the sensor observations given the target state and, therefore, should be compared to the proposed ReDif-PF algorithm in this paper in the known sensor parameters scenario of Section 4.1 as opposed to the more general Rao-Blackwellized version of ReDif-PF proposed for unknown sensor parameters in Section 4.2.

The main idea in MCDPF is to move each particle and its associated weight *multiple times* between nodes in the time interval between instants  $n$  and  $n+1$ , according to a Markov chain with transition probabilities defined by the normalized adjacency matrix  $\mathcal{A}$  of the graph  $\mathcal{G}$  that defines the network topology. Each time a given particle  $\mathbf{x}_n^*$  visits a network node  $r$ , its weight is multiplied by the pseudo-likelihood  $p(z_{r,n} | \mathbf{x}_n^*)^{1/J \phi(r)}$  where  $\phi(r)$  is the long-term stationary probability of the state of the Markov

chain specified by  $\mathcal{A}$  being equal to  $r$ ,  $r = 1, \dots, R$ , and  $J$  is total number of Markov chain move steps *between consecutive sensor measurements*, which is set by the user. Since the number of visits to the node  $r$  divided by  $J$  converges to  $\phi(r)$  [9] as  $J \rightarrow \infty$ , it follows that if  $J$  is large enough so that particle  $\mathbf{x}_n^*$  not only visits all network nodes but also visits each node multiple times, then the aggregate update factor for its corresponding weight at the end of the random walk will approach

$$\prod_{r=1}^R p(z_{r,n} | \mathbf{x}_n^*), \quad (38)$$

which, under the assumption of conditional independence of the sensor measurements given the target state, is the exact update factor for the optimal global weight associated with particle  $\mathbf{x}_n^*$ . For a finite and especially low number of move steps, MCDPF is no longer optimal, meaning that the choice of the parameter  $J$  involves a tradeoff between inter-node communication cost and state estimation error.

Contrary to MCDPF, the proposed ReDif-PF does not attempt to compute the exact optimal global posterior PDF  $p(\mathbf{x}_{0:n} | z_{1:R,0:n})$  at all nodes  $r = 1, \dots, R$  at each instant  $n$ . Instead, as explained in previous sections, ReDif-PF builds at each node  $r$  and at each instant  $n$  a Monte Carlo representation of the posterior  $p(\mathbf{x}_{0:n} | \mathcal{Z}_{r,0:n})$ , where  $\mathcal{Z}_{r,0:n}$  is a *random subset* of  $z_{1:R,0:n}$  that changes from node to node. Such Monte Carlo representation is built in a way that between instants  $n$  and  $n + 1$ , each node makes only *one request* to exchange particles/weights (or equivalent parametric approximations of posterior distributions) with a randomly chosen neighbor, thus eliminating the need for multiple iterative inter-node communication between consecutive sensor measurements and resulting in a communication cost that is much lower than that of the MCDPF algorithm for a similar mean square state estimation error (see the numerical simulation results in Section 5.2).

Finally, we also note that compared to the non-iterative ReDif-PF, MCDPF is also computationally more intensive since each node  $r$  has to compute the local likelihoods  $p(z_{r,n} | \mathbf{x}_n^{(q)})$  for all its particles  $\mathbf{x}_n^{(q)}$  multiple (namely  $J$ ) times between instants  $n$  and  $n + 1$ . We also illustrate that point in the numerical simulations of Section 5.2.

## 5 Simulation results

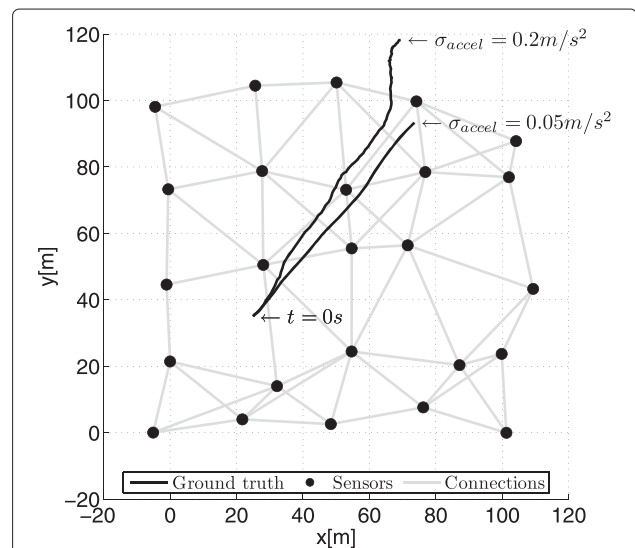
We assessed the performance of the proposed algorithms using 100 Monte Carlo runs with simulated data in three distinct scenarios assuming both unknown and known sensor variances. In all scenarios, we used  $R = 25$  RSS sensors with parameters  $P_0 = 1$  dBm,  $d_0 = 1$  m,  $\zeta_r = 3$ ,  $\forall r \in \mathcal{R}$ , and  $\sigma_r^2$  independently sampled at each node according to an  $\mathcal{IG}$  distribution with mean 16. The nodes

were deployed on a jittered grid within a square of size  $100 \text{ m} \times 100 \text{ m}$ . In the fully distributed algorithms, each node communicates with other nodes within a range of 40 m. All particle filters used  $Q = 500$  particles.

Figure 3 shows the sensor positions and two distinct realizations of the emitter trajectory generated for  $T = 1$  s and  $\mathbf{x}_0 = [25 \text{ m } 0.5 \text{ m/s } 35 \text{ m } 0.5 \text{ m/s}]^T$  considering respectively  $\sigma_{\text{accel}} = 0.05 \text{ m/s}^2$  and  $\sigma_{\text{accel}} = 0.2 \text{ m/s}^2$ . It also depicts the available network connections. The diameter of the sensor network is  $D = 5$  hops and the minimum number of neighbors for any possible node is 3.

### 5.1 Scenario I: ReDif-PF vs. CbPF

In the first scenario, we assumed unknown sensor variances and evaluated the performance of the Rao-Blackwellized ReDif-PF and two consensus-based PF trackers using respectively iterative minimum consensus (CbPFa) and flooding (CbPFb) (see also [11]). The aforementioned algorithms were compared to the equivalent broadcast implementation of the optimal centralized PF tracker, referred to as DcPF in [11] and [27] and in Section 3.1 of this paper. We also assumed Gaussian priors with mean  $[x_0 \ y_0]^T$  and covariance matrix  $\text{diag}(20^2, 20^2)$  for the emitter's position in Cartesian coordinates and mean  $[\sqrt{\dot{x}_0^2 + \dot{y}_0^2} \ \arctan(\dot{y}_0/\dot{x}_0)]^T$  and covariance matrix  $\text{diag}(0.3^2, (5\pi/180)^2)$  for the emitter's velocity in polar coordinates, where  $x_0 = 25 \text{ m}$ ,  $y_0 = 35 \text{ m}$ , and  $\dot{x}_0 = \dot{y}_0 = 0.5 \text{ m/s}$ . In the initialization step, the realizations of the initial emitter velocity are sampled from the



**Figure 3 Evaluated scenario.** Simulated scenario with  $R = 25$  RSS sensors deployed on a jittered grid within a square of size  $100 \text{ m} \times 100 \text{ m}$  and two distinct realizations of the emitter trajectory for  $\sigma_{\text{accel}} = 0.05 \text{ m/s}^2$  and  $\sigma_{\text{accel}} = 0.2 \text{ m/s}^2$ .

aforementioned Gaussian prior and, then, converted from polar to Cartesian coordinates.

Figure 4 shows the evolution of the root mean square (RMS) error norm - averaged over all network nodes and Monte Carlo runs - of the emitter position estimates for the RB ReDif-PF and the CbPFa and CbPFb algorithms superimposed to the benchmark RMS error curve for the optimal DcPF algorithm. Furthermore, we also show in Figure 4 the average RMS error norm for the non-cooperative (isolated) trackers and for a local cooperation scheme. In the former, each node runs a regularized PF tracker (see [11]) which assimilates local measurements only, while in the latter, a node  $r$  incorporates all measurements  $\mathcal{Z}_{r,n}$  in its vicinity in the same way as in the ReDif-PF tracker, but it does not exchange its updated posterior with its neighbors. The bars shown in Figure 4 represent the standard deviation of the error norm across all nodes in the network. There are no bars for the DcPF and CbPF algorithms since they provide the same state estimate at all nodes. The RMS error norm at time step 0 for all algorithms was calculated after the measurements  $z_{1:R,0}$  were assimilated. We implemented the RB ReDif-PF in this scenario with the parametric approximations in Section 4.3 using only one Gaussian mode to represent  $p(\mathbf{x}_{n-1}|\mathcal{Z}_{s,0:n-1})$ .

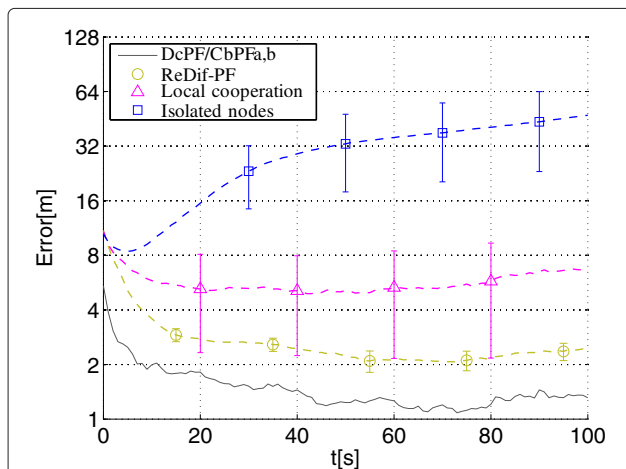
As expected, CbPFa and CbPFb match the performance of the DcPF tracker since both algorithms reproduce the optimal centralized PF tracker exactly, albeit with different communication and computational costs. On the other hand, as shown in Figure 4, the RB ReDif-PF tracker has a performance degradation compared to DcPF. This result is again theoretically expected since, in the RB ReDif-PF algorithm, the posterior at each node assimilates just a

subset of the available measurements  $z_{1:R,n}$  in the whole network at each time step  $n$ . However, ReDif-PF offers an improvement in error performance compared to the local cooperation scheme by better diffusing the information across the network. We also note from Figure 4 that the standard deviation of the state estimate across the different network nodes is much lower in the ReDif-PF algorithm than in the local cooperation scheme. Note also that, as shown in Figure 4, isolated nodes were not able to properly track the emitter in the evaluated scenario.

Finally, Figure 5 shows the performance comparison between the ReDif-PF and consensus-based algorithms for  $\sigma_{\text{accel}} \in \{0.05, 0.1, 0.2\}$ .

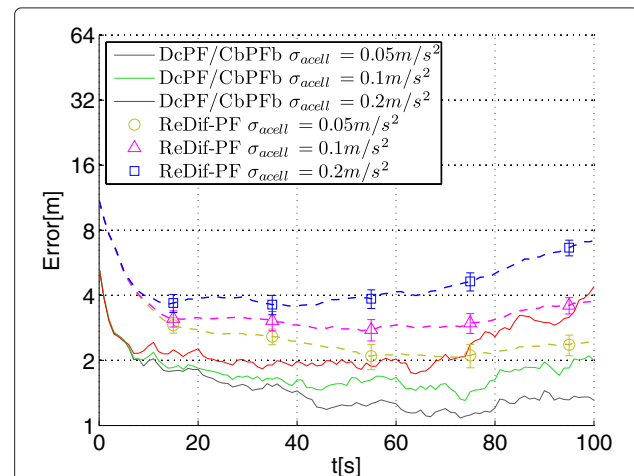
As expected, as  $\sigma_{\text{accel}}$  increases, there is a deterioration in the RMS error performance. However, the ratio between the RMS error performance of the sub-optimal ReDif-PF tracker and the benchmark optimal DcPF/CbPFb algorithms remains approximately constant (close to a factor of two) along the simulation period for all three different values of  $\sigma_{\text{accel}}$  employed.

**Communication and computation cost** Considering a four-byte and a one-byte network representation respectively for real and Boolean values, the total amount of bytes transmitted and received by all nodes over the network was recorded while running each tracker in Figure 4. Table 1 summarizes the communication cost for each algorithm in the first scenario (unknown sensor variances) in terms of average transmission (TX) and average reception (RX) rates per node and also quantifies the processing cost for each algorithm in terms of average duty cycle per node, measured in a Intel Core i5 machine with 4GB RAM. The duty cycle of a given node is defined as the



**Figure 4 Evolution of the estimated position RMS error norm.**

Performance comparison between the ReDif-PF and the consensus-based algorithms assuming a Gaussian prior distribution around the initial emitter state in the first scenario with *unknown* sensor noise variances.



**Figure 5 Evolution of the estimated position RMS error norm.**

Performance comparison between the ReDif-PF and the consensus-based algorithms for three different values of the parameter  $\sigma_{\text{accel}}$ .

**Table 1 Average communication and processing cost per node in the first scenario**

	RX rate	TX rate	Duty cycle (%)
DcPF	46.9 KB/s	2.0 KB/s	1.8
CbPFa	1.2 MB/s	244.1 KB/s	21.4
CbPFb	232.0 KB/s	48.7 KB/s	22.6
RB ReDif-PF	531.5 B/s	515.7 B/s	7.7
Local cooperation	4 B/s	19.8 B/s	9.3
Isolated nodes	-	-	2.0

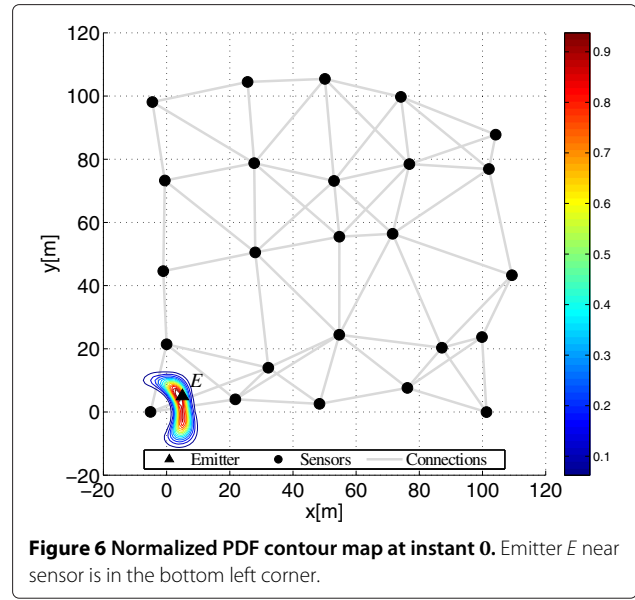
ratio between the total node processing time and the simulation period 100 s. Finally, values in Table 1 are averaged over all Monte Carlo simulations.

As shown in Table 1, the RB ReDif-PF tracker with the parametric approximations in Section 4.3 using only one Gaussian mode has a communication cost based on TX rate that is approximately one order of magnitude lower than the flooding-based CbPFb's communication requirements. Compared to the iterative minimum consensus solution (CbPFa), the average communication cost is reduced by two orders of magnitude.

## 5.2 Scenario II: ReDif-PF vs. ReDif-EKF

In the second scenario, the sensor variances are perfectly known and the ReDif-PF tracker is compared both to the optimal centralized PF and to a linearized random exchange extended Kalman filter (ReDif-EKF), which is summarized in Appendix 3. In the simulations, we assumed a non-informative prior for the sensor's initial position that is uniform in the entire surveillance space. The actual initial position of the emitter was, however, sampled from a Gaussian distribution centered at (5 m, 5 m) with standard deviation of 3 m in both dimensions. Figure 6 shows a normalized contour map for the posterior PDF  $p(x_0, y_0 | z_{1:R,0})$  at instant 0 as a function of  $x_0$  and  $y_0$  assuming the aforementioned non-informative prior. As seen from Figure 6, the initial posterior distribution of the target's position is non-Gaussian.

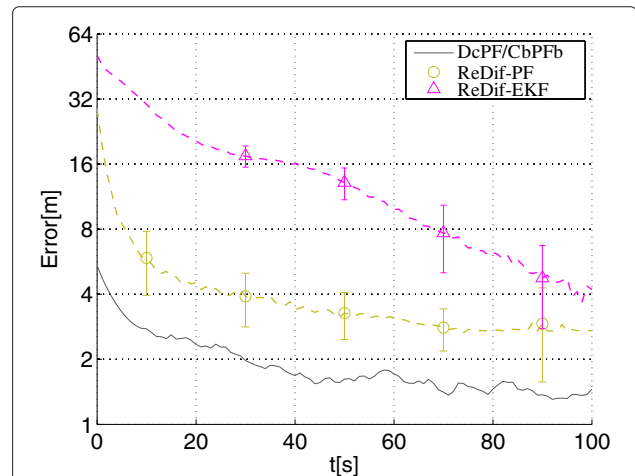
Figure 7 shows the evolution of the RMS error norm assuming known sensor variances respectively for the ReDif-PF algorithm in Section 4.1 with a two-Gaussian GMM parametric approximation and the ReDif-EKF algorithm in Appendix 3. We also show the RMS curve for the optimal centralized PF tracker as a benchmark. The plots in Figure 7 show that, especially in the initial time steps, when the posterior distribution of the states is strongly non-Gaussian as suggested by Figure 6, the fully distributed ReDif-PF outperforms its linearized counterpart, the ReDif-EKF. As the emitter moves away from the near field of the initial dominant sensor, the performance of the ReDif-EKF slowly improves and approaches that of the ReDif-PF, albeit still with a slight degradation towards the end of the simulation.



**Figure 6 Normalized PDF contour map at instant 0.** Emitter  $E$  near sensor is in the bottom left corner.

**Communication and Computation Cost** Table 2 summarizes the communication and processing cost per node for each algorithm in the second scenario.

As expected, the DcPF algorithm assuming known sensor variances has the same communication requirements as in the scenario with unknown variances since DcPF locally computes the likelihood functions and then broadcasts them to the entire network. However, as shown in Table 2, DcPF has a slightly lower processing cost when the sensor variances are known. The ReDif-PF tracker on the other hand outperformed the ReDif-EKF tracker in terms of the position RMS error at the expense of a greater communication and computational cost. However, as indicated in Table 2, the communication requirements



**Figure 7 Evolution of the estimated position RMS error norm.**

Performance comparison between the ReDif-PF and the ReDif-EKF algorithms assuming a non-informative prior in the second scenario with *known* sensor noise variances.

**Table 2 Average communication and processing cost per node in the second scenario**

	RX rate	TX rate	Duty cycle (%)
DcPF	46.9 KB/s	2.0 KB/s	0.8
ReDif-PF	643.4 B/s	627.6 B/s	15.1
ReDif-EKF	131.8 B/s	116.0 B/s	0.08

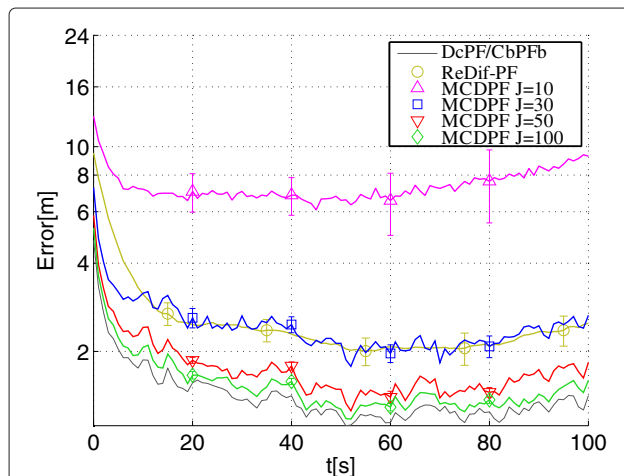
of the ReDif-PF and ReDif-EKF trackers still have the same order of magnitude.

### 5.3 Scenario III: ReDif-PF vs. MCDPF/selective gossip

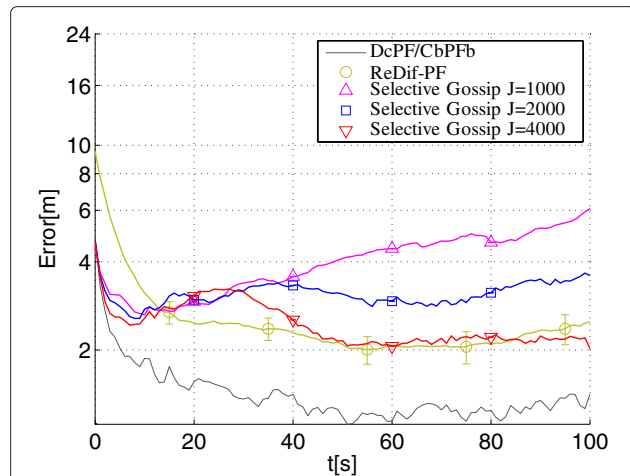
In the third scenario, the ReDif-PF tracker is compared to two iterative algorithms from the literature - the MCDPF and the selective gossip from [9] and [23], respectively - assuming perfectly known sensor variances as in the second scenario and the same Gaussian priors for the emitter's initial position and velocity used in the first scenario.

Figure 8 shows the evolution of the RMS error norm assuming known sensor variances for the ReDif-PF algorithm in Section 4.1 with a single-mode GMM parametric approximation and the MCDPF algorithm in [9] for  $J \in \{10, 30, 50, 100\}$  iterations.

Figure 9 shows the evolution of the RMS error norm for the ReDif-PF algorithm in Section 4.1 with a single-mode GMM parametric approximation and the selective gossip algorithm in [23] using respectively  $J \in \{1, 000; 2, 000; 4, 000\}$  iterations. More specifically, we first run  $J$  average gossip iterations considering only the particles in the top 10% bracket in terms of log-likelihood for each randomly selected pair of nodes at each iteration and, subsequently, we run  $J$  standard max gossip iterations



**Figure 8 Evolution of the estimated position RMS error norm.** Performance comparison between the ReDif-PF and the MCDPF algorithms assuming a Gaussian prior distribution around the initial emitter state in the third scenario with *known* sensor noise variances.



**Figure 9 Evolution of the estimated position RMS error norm.** Performance comparison between the ReDif-PF and the selective gossip algorithms assuming a Gaussian prior distribution around the initial emitter state in the third scenario with *known* sensor noise variances.

for the averaged log-likelihood of the selected particle as proposed in [23] to ensure that all nodes have exactly the same weight update factors. Note that, since only one pair of nodes is active at each average gossip iteration and only 10% of the particles are being transmitted between the active nodes, the Selective Gossip algorithm has a lower inter-node communication cost than MCDPF even when a much larger number of iterations is used between consecutive sensor measurements.

**Communication and computation cost** Table 3 summarizes the communication and processing cost per node for each algorithm in the third scenario.

The MCDPF and the selective gossip algorithms have a RMS error performance similar to the ReDif-PF algorithm for  $J = 30$  and  $J = 4,000$  iterations, respectively, at the expense of a communication cost approximately

**Table 3 Average communication and processing cost per node in the third scenario**

	RX rate	TX rate	Duty cycle (%)
DcPF	46.9 KB/s	2.0 KB/s	0.8
CbPFb	232.0 KB/s	48.7 KB/s	22.2
ReDif-PF	531.5 B/s	515.6 B/s	2.3
MCDPF $J = 10$	97.7 KB/s	97.7 KB/s	4.0
MCDPF $J = 30$	293.0 KB/s	293.0 KB/s	11.5
MCDPF $J = 50$	488.3 KB/s	488.3 KB/s	19.4
MCDPF $J = 100$	976.6 KB/s	976.6 KB/s	40.5
Selective gossip $J = 1,000$	31.3 KB/s	31.3 KB/s	4.3
Selective gossip $J = 2,000$	62.5 KB/s	62.5 KB/s	8.1
Selective gossip $J = 4,000$	125.0 KB/s	125.0 KB/s	15.8



two orders of magnitude larger than that of the ReDif-PF tracker. Moreover, for a comparable RMS error, the measured ReDif-PF duty cycle is also approximately five and seven times lower than the duty cycle of the MCDPF and the selective gossip algorithms respectively. Note, however, that the selective gossip tracker converges to the same estimate at all nodes and the estimates at each node provided by the MCDPF tracker have a lower standard deviation than those provided by the ReDif-PF algorithm.

We also note from Table 3 that with  $J = 100$  Markov chain move steps between sensor measurements, the MCDPF RMS error approaches the error curve of the optimal flooding-based CbPFb tracker with a inter-node communication cost that is, however, roughly four times greater than that of the CbPFb algorithm.

## 6 Conclusions

We introduced in this paper a Rao-Blackwellized version of the random exchange diffusion particle filter which enables fully distributed tracking of hidden state vectors in cooperative sensor networks with unknown sensor parameters. Although the general structure of the algorithm can be generalized to arbitrary signal models, we specified the algorithm in this particular paper in an application where we track a moving emitter using multiple RSS sensors with unknown noise variances. The ReDif-PF tracker, introduced originally in a simpler version in [17], is based on random information dissemination and is well suited for real-time applications since, unlike consensus-based approaches, it does not require iterative inter-node communication between measurement arrivals.

The new Rao-Blackwellized version of the ReDif-PF was compared to an exact broadcast implementation of the optimal centralized PF solution, referred to as the DcPF algorithm, and to two equivalent, fully distributed PFs using respectively iterative minimum consensus (CbPFa) and flooding (CbPFb). As expected, due to its suboptimality, the ReDif-PF tracker showed a degradation in RMS error performance compared to both DcPF and the equivalent consensus implementations in our simulations, but required much lower communication bandwidth with savings of one order of magnitude compared to DcPF and CbPFb in terms of transmission rate, and two orders of magnitude compared to CbPFa. The communication cost savings in the RB ReDif-PF algorithm were possible due to suitable parametric approximations introduced in Section 4.3.

The RB ReDif-PF algorithm RMS error performance was also compared in the unknown variance scenario to a local cooperation scheme in which each node assimilates all available measurements in its neighborhood but does not exchange its posterior statistics with other nodes. By diffusing information over the network, the RB ReDif-PF tracker showed better error performance than the

local cooperation scheme that uses local information only. Additionally, the standard deviation of the error norm considering all nodes in the network was much lower for RB ReDif-PF than in the local cooperation scheme, suggesting possible weak consensus.

Next, in a second scenario with perfectly known variances, we also compared a non-RB ReDif-PF tracker to its distributed linear filtering counterpart, the ReDif-EKF described in Appendix 3. Due to the non-Gaussianity of the posterior distribution of the states, the distributed PF solution outperformed the distributed EKF solution, albeit, as expected, at a greater computational and communication cost.

Finally, in a third scenario also with perfectly known variances, we compared the non-RB ReDif-PF tracker to two alternative distributed particle filters based respectively on iterative Markov chain move steps between sensor measurements as proposed in [9] and on iterative selective average gossiping as proposed in [23]. In our simulations, the novel ReDif-PF matched the RMS error performance with both the Markov chain and the selective gossip filters with an inter-node communication cost approximately two orders of magnitude lower and a required duty cycle that is reduced by a factor of 5 when compared to MCDPF and a factor of 7 when compared to the selective gossip scheme.

As future work, we plan to extend the ReDif-PF algorithm to perform joint detection and tracking-considering scenarios with probability of detection less than 1 and probability of false alarm greater than 0 as in [33]. We also plan to analyze the diffusion properties of ReDif-PF by investigating the long-term statistical properties of the sequence of visited nodes  $\{r_n\}$ ,  $n > 0$ , defined by the random exchange protocol starting from a random node  $r_0$ .

## Appendix 1

In this appendix, we use an importance sampling methodology (see [5,6]) to show that the augmented particle set  $\mathbf{x}_{r,0:n}^{(q)} = \{(\mathbf{x}_{s,0:n-1}^{(q)}, \mathbf{x}_{r,n}^{(q)})\}$ ,  $q = 1, \dots, Q$  with weights  $\{w_{r,n}^{(q)}\}$  obtained according to (12) and (13) in Section 4 is a properly weighted set to represent the posterior PDF  $p(\mathbf{x}_{0:n}|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1})$  in the sense that for any measurable function  $\mathbf{h}(\cdot)$ ,

$$E\{\mathbf{h}(\mathbf{x}_{0:n})|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}\} \approx \sum_{q=1}^Q w_{r,n}^{(q)} \mathbf{h}(\mathbf{x}_{r,0:n}^{(q)}).$$

Specifically, let  $\{\mathbf{x}_{s,0:n-1}^{(q)}\}$  with associated weights  $\{w_{s,n-1}^{(q)}\}$ ,  $q \in \mathcal{Q}$ , be a properly weighted set that represents the posterior PDF  $p(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})$  at node  $s$ . Assuming that the particle set  $\{\mathbf{x}_{s,0:n-1}^{(q)}\}$  was sampled according to some proposal importance function

$\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})$ , the proper weights  $\{w_{s,n-1}^{(q)}\}$  may be written as [5,6]

$$w_{s,n-1}^{(q)} = \frac{w(\mathbf{x}_{s,0:n-1}^{(q)})}{\sum_{j=1}^Q w(\mathbf{x}_{s,0:n-1}^{(j)})} \quad q \in \mathcal{Q}, \quad (39)$$

where

$$w(\mathbf{x}_{s,0:n-1}^{(q)}) = \frac{p(\mathbf{x}_{s,0:n-1}^{(q)}|\mathcal{Z}_{s,0:n-1})}{\pi(\mathbf{x}_{s,0:n-1}^{(q)}|\mathcal{Z}_{s,0:n-1})}.$$

Assume next that node  $s$  sends its particle set and weights to a neighboring node  $r$  that can access at instant  $n$  the measurements  $\mathcal{Z}_{r,n} = \{z_{r,n}\} \cup \{z_{i,n}\}_{i \in \mathcal{N}_r}$ . For any measurable function  $\mathbf{h}(\cdot)$ , we note that

$$\begin{aligned} E\{\mathbf{h}(\mathbf{x}_{0:n})|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}\} \\ = \frac{\int \mathbf{h}(\mathbf{x}_{0:n}) \frac{p(\mathbf{x}_{0:n}|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})} p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1}) d\mathbf{x}_{0:n}}{\int \frac{p(\mathbf{x}_{0:n}|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})} p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1}) d\mathbf{x}_{0:n}}. \end{aligned} \quad (40)$$

Sampling now at node  $r$  new particles  $\mathbf{x}_{r,n}^{(q)} \sim p(\mathbf{x}_n|\mathbf{x}_{s,n-1})$  and building the augmented particle trajectories  $\mathbf{x}_{r,0:n}^{(q)} = (\mathbf{x}_{s,0:n-1}^{(q)}, \mathbf{x}_{r,n}^{(q)}) \sim p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})$  the integral on the right-hand side of (40) can be approximated as

$$\begin{aligned} E\{\mathbf{h}(\mathbf{x}_{0:n})|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}\} &\approx \frac{\frac{1}{Q} \sum_{q=1}^Q \mathbf{h}(\mathbf{x}_{r,0:n}^{(q)}) w(\mathbf{x}_{r,0:n}^{(q)})}{\frac{1}{Q} \sum_{j=1}^Q w(\mathbf{x}_{r,0:n}^{(j)})} \\ &= \sum_{q=1}^Q w_{r,n}^{(q)} \mathbf{h}(\mathbf{x}_{r,0:n}^{(q)}) \end{aligned} \quad (41)$$

where

$$w_{r,n}^{(q)} = \frac{w(\mathbf{x}_{r,0:n}^{(q)})}{\sum_{j=1}^Q w(\mathbf{x}_{r,0:n}^{(j)})} \quad (42)$$

and

$$\begin{aligned} w(\mathbf{x}_{0:n}) &= \frac{p(\mathbf{x}_{0:n}|\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_n|\mathbf{x}_{n-1})\pi(\mathbf{x}_{0:n-1}|\mathcal{Z}_{s,0:n-1})} \\ &= \frac{p(\mathcal{Z}_{r,n}|\mathbf{x}_{0:n}, \mathcal{Z}_{s,0:n-1}) p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathcal{Z}_{r,n}|\mathcal{Z}_{s,0:n-1})} \\ &\quad \times w(\mathbf{x}_{0:n-1}). \end{aligned} \quad (43)$$

Substituting (43) into (42) and recalling from the model assumptions that  $p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathcal{Z}_{s,0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$  we get the recursion

$$\begin{aligned} w_{r,n}^{(q)} &= \frac{w(\mathbf{x}_{s,0:n-1}^{(q)})p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1})}{\sum_{j=1}^Q w(\mathbf{x}_{s,0:n-1}^{(j)})p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(j)}, \mathcal{Z}_{s,0:n-1})} \\ &\propto \frac{w_{s,n-1}^{(q)}p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(q)}, \mathcal{Z}_{s,0:n-1})}{\sum_{j=1}^Q w_{s,n-1}^{(j)}p(\mathcal{Z}_{r,n}|\mathbf{x}_{r,0:n}^{(j)}, \mathcal{Z}_{s,0:n-1})} \end{aligned} \quad (44)$$

where, as before,

$$w_{s,n-1}^{(q)} = \frac{w(\mathbf{x}_{s,0:n-1}^{(q)})}{\sum_{j=1}^Q w(\mathbf{x}_{s,0:n-1}^{(j)})}.$$

## Appendix 2

Let, as before,

$$IG(\sigma^2|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \sigma^{-2(\alpha+1)} \exp\left(-\frac{\beta}{\sigma^2}\right) \quad (45)$$

$$N(z|m, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(z-m)^2}{2\sigma^2}\right] \quad (46)$$

where  $\sigma^2 > 0$  and  $m \in \Re$ . After some algebraic calculations, it can be shown (see [19] and also [6,11]) that

$$\frac{N(z|m, \sigma^2) IG(\sigma^2|\alpha, \beta)}{\int_0^\infty N(z|m, \sigma^2) IG(\sigma^2|\alpha, \beta) d\sigma^2} = IG(\sigma^2|\bar{\alpha}, \bar{\beta}) \quad (47)$$

where

$$\bar{\alpha} = \alpha + \frac{1}{2} \quad (48)$$

$$\bar{\beta} = \beta + \frac{1}{2} (z-m)^2. \quad (49)$$

Similarly, using the same algebraic procedure, it follows that (see [6,19])

$$\int_0^\infty N(z|m, \sigma^2) IG(\sigma^2|\alpha, \beta) d\sigma^2 \propto \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\bar{\alpha})}{\bar{\beta}^{\bar{\alpha}}} \quad (50)$$

where  $\bar{\alpha}$  and  $\bar{\beta}$  are given respectively by (48) and (49).

Assume now that at node  $s$  at instant  $n-1$ , the joint posterior PDF  $p(\sigma_{1:R}^2|\mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})$  is factored as

$$p(\sigma_{1:R}^2|\mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) = \prod_{i=1}^R IG(\sigma_i^2|\alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}). \quad (51)$$

In the sequel, assume that node  $s$  transmits to a neighboring node  $r$  its weighted particle set  $\{(w_{s,n-1}^{(q)}, \mathbf{x}_{s,n-1}^{(q)})\}$

and the corresponding parameters  $\{\alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}\}$ ,  $q = 1, \dots, Q$ ,  $i = 1, \dots, R$ . At instant  $n$ , as explained in Section 4, node  $r$  samples a new set of particles

$$\mathbf{x}_{r,n}^{(q)} \sim p(\mathbf{x}_n | \mathbf{x}_{s,n-1}^{(q)}) \quad (52)$$

and updates its weights as

$$\begin{aligned} w_{r,n}^{(q)} &= w_{s,n-1}^{(q)} p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) \\ &= w_{s,n-1}^{(q)} \int_0^\infty \dots \int_0^\infty \left[ p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_{1:R}^2) \right. \\ &\quad \times \left. p(\sigma_{1:R}^2 | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) \right] d\sigma_{1:R}^2 \\ &= \left\{ \prod_{i \in \tilde{\mathbf{N}}_r} \int_0^\infty \left[ p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) \right. \right. \\ &\quad \times \left. \left. IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) \right] d\sigma_i^2 \right\} \\ &\quad \times \prod_{i \notin \tilde{\mathbf{N}}_r} \underbrace{\int_0^\infty IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) d\sigma_i^2}_{=1}, \end{aligned} \quad (53)$$

where  $\tilde{\mathbf{N}}_r$  denotes  $\{r\} \cup \mathbf{N}_r$  and, as in Section 4,  $\mathcal{Z}_{r,n}$  is a notation for the set  $\{z_{i,n}\}$  for all  $i \in \tilde{\mathbf{N}}_r$ . In (53), we used the facts that

$$\begin{aligned} p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \sigma_{1:R}^2, \mathcal{Z}_{s,0:n-1}) &= p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_{1:R}^2) \\ &= \prod_{i \in \{r\} \cup \mathbf{N}_r} p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) \end{aligned} \quad (54)$$

and

$$\begin{aligned} p(\sigma_{1:R}^2 | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) &= \frac{p(\mathbf{x}_{r,n}^{(q)} | \mathbf{x}_{s,0:n-1}^{(q)}, \sigma_{1:R}^2, \mathcal{Z}_{s,0:n-1}) p(\sigma_{1:R}^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_{r,n}^{(q)} | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})} \\ &= \frac{p(\mathbf{x}_{r,n}^{(q)} | \mathbf{x}_{s,n-1}^{(q)}) p(\sigma_{1:R}^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1})}{p(\mathbf{x}_{r,n}^{(q)} | \mathbf{x}_{s,n-1}^{(q)})} \\ &= p(\sigma_{1:R}^2 | \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}), \end{aligned}$$

which, in turn, is assumed to be factored as in (51). On the other hand, using (50), it follows that for each  $i \in \{r\} \cup \mathbf{N}_r$ ,

$$\begin{aligned} &\int p(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) d\sigma_i^2 \\ &= \int N(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) d\sigma_i^2 \\ &\propto \frac{[\beta_{s,i,n-1}^{(q)}]^{\alpha_{s,i,n-1}}}{\Gamma(\alpha_{s,i,n-1})} \frac{\Gamma(\alpha_{r,i,n})}{[\beta_{r,i,n}^{(q)}]^{\alpha_{r,i,n}}} \end{aligned}$$

where, from (48) and (49),  $\alpha_{r,i,n}$  and  $\beta_{r,i,n}^{(q)}$  are given by (20) and (21) in Section 4.2.

Similarly, node  $r$  at instant  $n$  updates the posterior PDF of the unknown variances as

$$\begin{aligned} p(\sigma_{1:R}^2 | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}) &= C_n p(\mathcal{Z}_{r,n} | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \sigma_{1:R}^2, \mathcal{Z}_{s,0:n-1}) \\ &\quad \times p(\sigma_{1:R}^2 | \mathbf{x}_{r,n}^{(q)}, \mathbf{x}_{s,0:n-1}^{(q)}, \mathcal{Z}_{s,0:n-1}) \\ &= C_n \left[ \prod_{i \in \tilde{\mathbf{N}}_r} N(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) \right] \\ &\quad \times \left[ \prod_{i \notin \tilde{\mathbf{N}}_r} IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) \right] \\ &= \prod_{i=1}^R IG(\sigma_i^2 | \alpha_{r,i,n}, \beta_{r,i,n}^{(q)}) \end{aligned}$$

where

$$C_n = \left\{ \prod_{i \in \tilde{\mathbf{N}}_r} \left[ \int_0^\infty N(z_{i,n} | \mathbf{x}_{r,n}^{(q)}, \sigma_i^2) IG(\sigma_i^2 | \alpha_{s,i,n-1}, \beta_{s,i,n-1}^{(q)}) d\sigma_i^2 \right] \right\}^{-1}$$

is a normalization constant that does not depend on  $\sigma_{1:R}^2$  and, using (47), (48), and (49), for all  $i \in \{r\} \cup \mathbf{N}_r$ ,

$$\alpha_{r,i,n} = \alpha_{s,i,n-1} + \frac{1}{2} \quad (55)$$

$$\beta_{r,i,n}^{(q)} = \beta_{s,i,n-1}^{(q)} + \frac{1}{2} [z_{i,n} - g_i(\mathbf{x}_{i,n}^{(q)})]^2, \quad (56)$$

as in (20) and (21) in Section 4.2. Otherwise, if  $i \notin \{r\} \cup \mathbf{N}_r$ , then

$$\alpha_{r,i,n} = \alpha_{s,i,n-1} \quad (57)$$

$$\beta_{r,i,n}^{(q)} = \beta_{s,i,n-1}^{(q)}. \quad (58)$$

### Appendix 3

In a scenario with perfectly known sensor model parameters, assume that at instant  $n - 1$ , node  $s$  has a *linear*



estimate  $\hat{\mathbf{x}}_{s,n-1|n-1}$  of the hidden state  $\mathbf{x}_{n-1}$  based on the observations  $\mathcal{Z}_{s,0:n-1}$ , which were assimilated by node  $s$  from instant zero up to instant  $n-1$ .

In the sequel, as proposed in [2], assume that node  $s$  and a randomly chosen node  $r$  in the neighborhood of  $s$  exchange their respective estimates  $\hat{\mathbf{x}}_{s,n-1|n-1}$  and  $\hat{\mathbf{x}}_{r,n-1|n-1}$ , and the respective associated conditional covariance matrices,  $\mathbf{P}_{s,n-1|n-1}$  and  $\mathbf{P}_{r,n-1|n-1}$ .

At instant  $n$  then, we may get a new linear estimate  $\hat{\mathbf{x}}_{r,n|n}$  at node  $r$ , with associated conditional covariance matrix  $\mathbf{P}_{r,n|n}$ , propagating  $\hat{\mathbf{x}}_{s,n-1|n-1}$  and  $\mathbf{P}_{s,n-1|n-1}$  using the usual extended Kalman filter recursions, but assimilating now only the local measurements  $\{z_{i,n}\}$ ,  $i \in \{r\} \cup \mathbf{N}_r$ , also denoted  $\mathcal{Z}_{r,n}$ . Under that approach,  $\hat{\mathbf{x}}_{r,n|n}$  is now an approximate linear minimum mean square error estimate (see [6]) of the hidden state  $\mathbf{x}_n$  at instant  $n$  given the new set of observations  $\mathcal{Z}_{r,0:n} = \{\mathcal{Z}_{r,n}, \mathcal{Z}_{s,0:n-1}\}$ .

Specifically, for a more general state-space model of the form

$$\mathbf{x}_{n+1} = \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{u}_n \quad n \geq 0 \quad (59)$$

$$\mathbf{z}_{r,n} = \mathbf{h}_r(\mathbf{x}_n) + \mathbf{v}_{r,n} \quad n \geq 0, r = 1, \dots, R \quad (60)$$

with  $E\{\mathbf{u}_n \mathbf{u}_n^T\} = \mathbf{Q}_n$  and  $E\{\mathbf{v}_{r,n} \mathbf{v}_{r,n}^T\} = \mathbf{R}_{r,n}$ , the prediction step of the extended Kalman filter at node  $r$  at instant  $n$ , after parameter exchange, is given by

$$\hat{\mathbf{x}}_{r,n|n-1} = \mathbf{F}_{n-1} \hat{\mathbf{x}}_{s,n-1|n-1}.$$

$$\mathbf{P}_{r,n|n-1} = \mathbf{F}_{n-1} \mathbf{P}_{s,n|n-1} \mathbf{F}_{n-1}^T + \mathbf{G}_{n-1} \mathbf{Q}_{n-1} \mathbf{G}_{n-1}^T.$$

On the other hand, making

$$\mathbf{H}_{i,n} = \frac{\partial \mathbf{h}_i(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{r,n|n-1}} \quad i \in \{r\} \cup \mathbf{N}_r,$$

the updated step equations of the distributed EKF become

$$(\mathbf{P}_{r,n|n})^{-1} = (\mathbf{P}_{r,n|n-1})^{-1} + \sum_{i \in \{r\} \cup \mathbf{N}_r} \mathbf{H}_{i,n}^T \mathbf{R}_{i,n}^{-1} \mathbf{H}_{i,n}.$$

$$\begin{aligned} \hat{\mathbf{x}}_{r,n|n} &= \hat{\mathbf{x}}_{r,n|n-1} + \mathbf{P}_{r,n|n} \sum_{i \in \{r\} \cup \mathbf{N}_r} \left[ \mathbf{H}_{i,n}^T \mathbf{R}_{i,n}^{-1} \right. \\ &\quad \left. \times (\mathbf{z}_{i,n} - \mathbf{H}_{i,n} \hat{\mathbf{x}}_{r,n|n-1}) \right]. \end{aligned}$$

Note that in the updated step of the random exchange distributed EKF, node  $r$  must have access to the measurements  $\{z_{i,n}\}$  from its immediate neighbors and must also know their respective sensor covariance matrices  $\{\mathbf{R}_{i,n}\}$  and the analytic expressions of the neighboring gradients  $\left\{ \frac{\partial \mathbf{h}_i(\cdot)}{\partial \mathbf{x}} \right\}$ , which are then all evaluated locally at node  $r$  at the predicted estimate  $\hat{\mathbf{x}}_{r,n|n-1}$ . Alternatively, node  $r$  may

transmit  $\hat{\mathbf{x}}_{r,n|n-1}$  to its neighbors, which then evaluate their respective gradients and transmit back the matrices  $\{\mathbf{H}_{i,n}\}$  and  $\{\mathbf{R}_{i,n}\}$  to node  $r$ .

#### Competing interests

The authors declare that they have no competing interests.

#### Acknowledgements

The authors would like to thank Professor José M. F. Moura for fruitful discussions at ICASSP 2012 that motivated this work. The authors would also like to acknowledge Dr. Claudio Bordin Jr. for helpful discussions on the topic of inter-node communication cost in network particle filtering.

#### Author details

<sup>1</sup>Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes 50, São José dos Campos, São Paulo 12228-900, Brazil. <sup>2</sup>Embraer Defense & Security, Av. Brigadeiro Faria Lima 2.170, São José dos Campos, São Paulo 12227-901, Brazil.

Received: 1 October 2013 Accepted: 24 January 2014

Published: 13 February 2014

#### References

1. PM Djurić, J Beaudreau, MF Bugallo, Non-centralized target tracking with mobile agents, in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Prague, 2011), pp. 5928–5931
2. S Kar, JMF Moura, Gossip and distributed Kalman filtering: weak consensus under weak detectability. *IEEE Trans. Signal Process.* **59**(4), 1766–1784 (2011)
3. FS Cattivelli, AH Sayed, Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Trans. Automatic Control.* **55**(9), 2069–2084 (2010)
4. A Ribeiro, GB Giannakis, SI Roumeliotis, SOL-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Trans. on Signal Process.* **54**(12), 4782–4795 (2006)
5. A Doucet, S Godsill, C Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10**(3), 197–208 (2000)
6. MGS Bruno, Sequential Monte Carlo methods for nonlinear discrete-time filtering. *Synth. Lect. Signal Process.* **6**(1), 1–99 (2013)
7. O Hlinka, F Hlawatsch, PM Djurić, Distributed particle filtering in agent networks: a survey, classification, and comparison. *IEEE Signal Process. Mag.* **30**(1), 61–81 (2013)
8. O Hlinka, O Sluciak, F Hlawatsch, PM Djurić, M Rupp, Likelihood consensus and its application to distributed particle filtering. *IEEE Trans. Signal Process.* **60**(8), 4334–4349 (2012)
9. SH Lee, M West, Markov chain distributed particle filters (MCDPF), in *Proceedings of the 48th IEEE International Conference on Decision and Control* (IEEE, Shanghai, 2009), pp. 5496–5501
10. D Ustebay, M Coates, M Rabbat, Distributed auxiliary particle filters using selective gossip, in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Prague, 2011), pp. 3296–3299
11. SS Dias, MGS Bruno, Cooperative target tracking using decentralized particle filtering and RSS sensors. *IEEE Trans. Signal Process.* **61**(14), 32–3646 (2013)
12. V Yadav, MV Salapaka, Distributed protocol for determining when averaging consensus is reached, in *45th Annual Allerton Conference*, (Allerton House - UIUC, 2007), pp. 715–720
13. D Tsoumakos, N Roussopoulos, A comparison of peer-to-peer search methods, in *Proceedings of the WebDB* (Citeseer, San Diego, 2013), pp. 61–66
14. S Farahmand, SI Roumeliotis, GB Giannakis Particle filter adaptation for distributed sensors via set membership, in *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (IEEE, Dallas, 2010), pp. 3374–3377
15. A Mohammadi, A Asif, Consensus-based distributed unscented particle filter, in *Proceedings of the 2011 IEEE Statistical Signal Processing Workshop (SSP)* (IEEE, Nice, 2011), pp. 237–240
16. AH Sayed, S-Y Tu, J Chen, X Zhao, ZJ Towfic, Diffusion strategies for adaptation and learning over networks: An examination of distributed

- strategies and network behavior. *IEEE Signal Process. Mag.* **30**(3), 155–171 (2013)
17. SS Dias, MGS Bruno, Distributed emitter tracking using random exchange diffusion particle filters, in *Proceedings of the 16th International Conference on Information Fusion* (IEEE, Istanbul, 2013)
  18. G Casella, CP Robert, Rao-blackwellisation of sampling schemes. *Biometrika*. **83**(1), 81–94 (1996)
  19. A Gelman, JB Carlin, HS Stern, DB Rubin, *Texts in Statistical Science - Bayesian Data Analysis*, 2nd edn. (Chapman & Hall/CRC, Florida, 2003)
  20. SS Dias, MGS Bruno, A Rao-Blackwellized random exchange diffusion particle filter for distributed emitter tracking, in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (IEEE, St. Martin, 2013)
  21. X Sheng, Y-H Hu, P Ramanathan, Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network, in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05* (IEEE Press, Los Angeles, 2005), pp. 181–188
  22. CJ Bordin, MGS Bruno, A particle filtering algorithm for cooperative blind equalization using VB parametric approximations, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (IEEE, Dallas, 2010), pp. 3834–3837
  23. D Üstebay, R Castro, M Rabbat, Selective gossip, in *Proceedings of the 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (IEEE, Aruba, 2009), pp. 61–64
  24. Y Bar-Shalom, X-R Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. (University of Connecticut, Storrs, CT, 1995)
  25. N Patwari, AO Hero III, M Perkins, NS Correal, RJ O'dea, Relative location estimation in wireless sensor networks. *Proc. IEEE Trans. Signal Process.* **51**(8), 2137–2148 (2003)
  26. M Coates, Distributed particle filters for sensor networks, in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks* (ACM, New York, 2004), pp. 99–107
  27. CJ Bordin, MGS Bruno, Cooperative blind equalization of frequency-selective channels in sensor networks using decentralized particle filtering, in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers* (IEEE, Pacific Grove, 2008), pp. 1198–1201
  28. L Xiao, S Boyd, Fast linear iterations for distributed averaging. *Syst. & Control Lett.* **53**(1), 65–78 (2004)
  29. CJ Bordin, MGS Bruno, Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks, in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Prague, 2011), pp. 3968–3971
  30. S Boyd, A Ghosh, B Prabhakar, D Shah, Randomized gossip algorithms. *IEEE Trans. Inf. Theory*. **52**(6), 2508–2530 (2006)
  31. AP Dempster, NM Laird, DB Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.* **39**(1), 1–38 (1977)
  32. SS Dias, MGS Bruno, Cooperative particle filtering for emitter tracking with unknown noise variance, in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Kyoto, 2012), pp. 2629–2632
  33. MGS Bruno, RV Araújo, AG Pavlov, Sequential monte carlo methods for joint detection and tracking of multiaspect targets in infrared radar images. *EURASIP J. Adv. Signal Process.* **2008** (2008). doi:10.1155/2008/217373

doi:10.1186/1687-6180-2014-19

**Cite this article as:** Bruno and Dias: Collaborative emitter tracking using Rao-Blackwellized random exchange diffusion particle filtering. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:19.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)